The Victorian 6502 User Group Newsletter

KAOS

For People Who Have Got Smart

HARDWARE: David Anear SOFTWARE: Jeff Rae PASCAL: Tony Durrant
NEWSLETTER: Ian Eyles EDUCATION: Mike Pekin
AMATEUR RADIO: Rod Drysdale VK3BYU

ADDRESS ALL CORRESPONDENCE TO 10 FORBES ST., ESSENDON, VIC. 3040

Registered for posting as
a publication Category B

Vol:.2 No.3

December 1981

A belated Merry Christmas to you all. Because of the postal dispute and the fact that the newsletter is sent bulk postage (which means that it wouldn't be handled till the back-log of mail was cleared) there was not much point in sending the newsletter earlier. In fact we probably would not have sent it at all except we have more information about Australian Source which we feel you should get as early as possible so that you can make up your minds before the February 1st deadline. See the article on page 3

We have a couple of wrong mailing addresses. If anyone knows J. Phillips of Caulfield Sth or Wal Mills of St Kilda Rd, Melbourne would they please ask them to advise us of their correct address. We are at the present revising our mailing list so if there are any errors in the address on your newsletter, please let us know. Just a note on your membership renewal form will do. By the way, remember that membership fees are due in January 1982.

Bruce Fisher of LOOKY VIDEO has advised us that from January 1st, his shop in Richmond will be closed and he will be operating LOOKY VIDEO from his home as a mail and phone order business. LOOKY VIDEO's new address is P.O. BOX 254, HEIDELBERG, VICTORIA 3084 and the phone number is (03) 435 3004. Bruce plans to be available to answer phone queries between 1pm and 6pm Monday to Saturday.

A new batch of KAOS T-shirts has arrived. In case you don't know, they are a white T-shirt with the words GET SMART JOIN KAOS on the front. They are available in sizes 6 to 22 at $4.95 each plus 50 cents postage.

We have arranged to continue using the Teachers' Centre at Essendon Primary School for our meetings in 1982. The next meeting will be at the school at 2pm on Sunday 31st January 1982. As the children will still be on holidays and the principal has to come in especially to open up for us, the school will not be open till 1pm.

WE WOULD LIKE TO WISH YOU ALL A VERY HAPPY AND PROSPEROUS NEW YEAR.

THE MEETING WAS KAOS

VORTEX:- Gerry demonstrated his speech synthesiser called Vortex, which is made by Atari but can be fitted to most computers and costs about $375.00 us. It is far more flexible than Digi-Talker as it does not use words that are in ROM but actually contructs them under instruction from the computer. Then George took over the controls and showed us how easy it was to operate and words like EXTERMINATE and KAOS certainly made it a humorous meeting. One use that comes to mind is as an automatic call sign identification for the Radio Amateur Service. I recently saw an Amateur Television operator using the Olinda ATV repeater using Digi-Talker for automatic identification. I wonder if they will produce speech synthesisers with Australian accents in future.

DAC:- (David Anear Club) Hi! Hi! David Anear played the Star Wars theme using his 8 bit Digital to Anolog Converter on his C8P. At present a number of tunes are available on 8" disk, though due to high interest, we hope they will soon be available on 5 1/4" disk for those budding musicians with C1P's series II.

JOYSTICKS:- New tilt type joysticks are available from Atari. They use mercury switches and could easily be made to suit the Superboard. I believe mercury switches are available through Tandy.

GTBUG:- Tony called on six volunteers to fully test his new 2K monitor. What is the old saying about the quick and the dead? Who knows, but I nearly got crushed to death in the rush to reach Tony. I have had no feed-back at present on the progress of the 2K monitors, we will probably hear early in 1982.

AUSTRALIAN SOURCE:- Ian and Jeff have been investigating the Australian Source on behalf of KAOS. The group discussed fees and services of Australian Source and there is more news on this in the newsletter.

SYM:- The hardware committee will be looking in some detail at using the Tasker Buss system for the SYM in 1982.

LOOKY VIDEO:- Bruce Fisher of Looky Video indicated that in 1982 they will be going mail order only.

Well that was the November meeting, short and sweet. The past twelve months have proved to be an exciting period of expansion which was far greater than anyones expectations. The total membership for 1981 is 293, an unbelievable for a club only fifteen months old. We can only hope that 1982 will be just as successful.

73's Rod Drysdale
VK3BYU

0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0

TELETYPE

To use the TELETYPE routine with DABUG III, change the 1st instruction from JSR FF69 to JSR FB00.

If you use the Extended Monitor to printout disassembled code, change 099D from 17 to 46 to print one A4 page before stopping.

John Whitehead

THE AUSTRALIAN SOURCE

Last week Jeff Rae and I met Garry Albert from the Australian Source and had a very interesting discussion which cleared up a few points about which club members have been asking. I will answer the most asked question first.

1. OSI Computers are now on the 'approved' computer list for the Australian Source.

2. OSI programs will be in the data banks when the computer goes into operation in February, all non-commercial programs in the KAOS library will be on the Source as soon as possible and new programs will be entered as soon as they are received.

3. Everything advertised in the brochure will be available in February with the exception of the Electronic Mailbox, which is still awaiting Telecom approval.

4. The first system in operation will be Melbourne with Sydney, Brisbane, Perth and Adelaide starting, in that order, at 60 day intervals. There will be a few 08 (toll free) lines available but these will be hard to get on to, so if you want to use the Source before your capital city comes on line, you will have to either call in the middle of the night, or make an STD call to Melbourne. The Source will be going onto Austpac as soon as it is available. (Telecom informs us this will be late 1982.) This will mean the Source will be available from anywhere in Australia for a local call fee.

5. There is a once only joining fee and no monthly or minimum charge. If you are enrolled before 1st February as a member of a user group, your joining fee will be $40.00 and you will be entitled to the 20% discount on user charges. If you join as a user group member after the initial period, your joining fee will be around $65.00.

To make it easier for the people at the Source, we have agreed to make the initial membership applications in a block, so if you are interested in joining, send your name and address and a cheque etc. for $40.00 (payable to KAOS, or if you prefer, The Australian Source) to KAOS, 10 Forbes st, Essendon, Victoria 3040. We will register you with the Source and you will be mailed an instruction manual and an application form, on which you list type of computer etc.. You send this form back to the Source and they will then post you your operating manual and your personal code number. These will be sent by registered post, so there is no chance of anyone but you and the Source knowing your personal code number.

If you wish to make your own application, please mark your form "KAOS USER GROUP".

The $40.00 joining fee only applies if 20 or more members join, but judging by the number of enquiries we have had, we will get many more than this. If we enroll more than 50 before the 1st of February, the joining fee goes down to $35.00. If this occurs, we will either refund the $5.00 to you or arrange with the Source to give you a credit of $5.00 user time.

SYM-POSIUM

A few years ago Synertek Systems Corporation introduced a single board microcomputer known as the VIM-1 (which stood for 'Versatile Interface Module'). Shortly after its release it was renamed the SYM.

Those of you who own one will realize just how accurate the first name was. The SYM is an extremely versatile system. Unfortunately, the SYM Reference Manual only covers the more fundamentals of the microcomputer.

This is the first article of a series which will go deeper into the SYM's capabilities, elaborating on the details given in the manual and in many cases the articles will cover points not even mentioned in the manual. The articles will cover topics such as the monitor program (SUPERMON), the system RAM, tape I/O, user RESET routines, monitor extensions, the debugger and much more. But first:-

THE ROM SOCKETS

The four ROM sockets on your board are capable of taking a variety of ROM or EPROM devices. The manual gives the strappings for the 2716, 2316, 2332 and the 2364. Since the printing of the manual, the 2732 EPROM has dropped in price and is now as economic as the 2716. Shown below are the strappings required for 2732 EPROMs inserted in P0 to P3.

SOCKET LOCATION	SOCKET NAME	MEMORY DEVICE	JUMPER LETTER	POSITION NUMBER
U20	P0	2732	A	2 or 3
			E	1
			J	*
U21	P1	2732	B	2 or 3
			F	1
			K	*
U22	P2	2732	C	2 or 3
			G	1
			L	*
U23	P3	2732	D	2 or 3
			H	1
			M	*

* This pin addresses the EPROM

CONNECT TO	FOR ADDRESS	
7 and 8	$8000-$8FFF	SUPERMON
9 and 10	$9000-$9FFF	
46 and 47 (may be unlabeled)	$B000-$BFFF	RAE
11 and 12	$C000-$CFFF	BASIC
13 and 14	$D000-$DFFF	BASIC
15 and 16	$E000-$EFFF	RAE
17 and 18	$F000-$FFFF	SYSTEM RAM (ALIAS)

It is important to note that P0 is slightly different to the other sockets, P1-P3 in that P0 is connected to the debugger circuit. When a routine is executed, the debugger is disabled so that the routine is executed at the normal speed. SUPERMON is currently in this socket and should remain since it contains the terminal I/O routines which contain timing loops.

Next month - USER RESET ROUTINES

Would you like your SYM to run BASIC the moment you power up? Or maybe you have a program of your own you would like it to execute when you turn your SYM on. The SYM allows the user to implement his own RESET routines very easily.

Also an EPROM programmer capable of programming 2716 2732 EPROMs which will cost $15 or less is nearly complete. With any luck the circuit diagram and software will be available next month.

Brian Campbell

****

ATARI NEWS

AUSTRALIA'S FIRST ATARI USER'S GROUP M. A. C. E.

Welcome to the Christmas edition of Atari news. Hope your hangovers were not too bad. Futuretronics have loaned us the much heralded 16K Microsoft Basic, at the moment it is on disk. For some strange reason it requires 32K of memory to operate. It has very good documentation and is well written and presented. In the front of the manual is a section showing how tp convert other basic programs to Atari Microsoft.

A surprise announcement from Futuretronics. There is a Microsoft version of Pascal. It will be over here in January. I don't know what format it will be on yet, I will let you know when I find out.

PACMAN (Ghost Munchers) has had a forced withdrawal from the market. Atari took the offending company to court and won. You will have to get your hands on a back-up copy if you want to be eaten by Ghosts, or wait for Atari to bring their version out.

Futuretronics have a few sets of Technical Manuals, contact John Lemic. Price on application. I have just received the Monkey Wrench ROM I talked about last month, it is a very good program. Let me know if you want to use it.

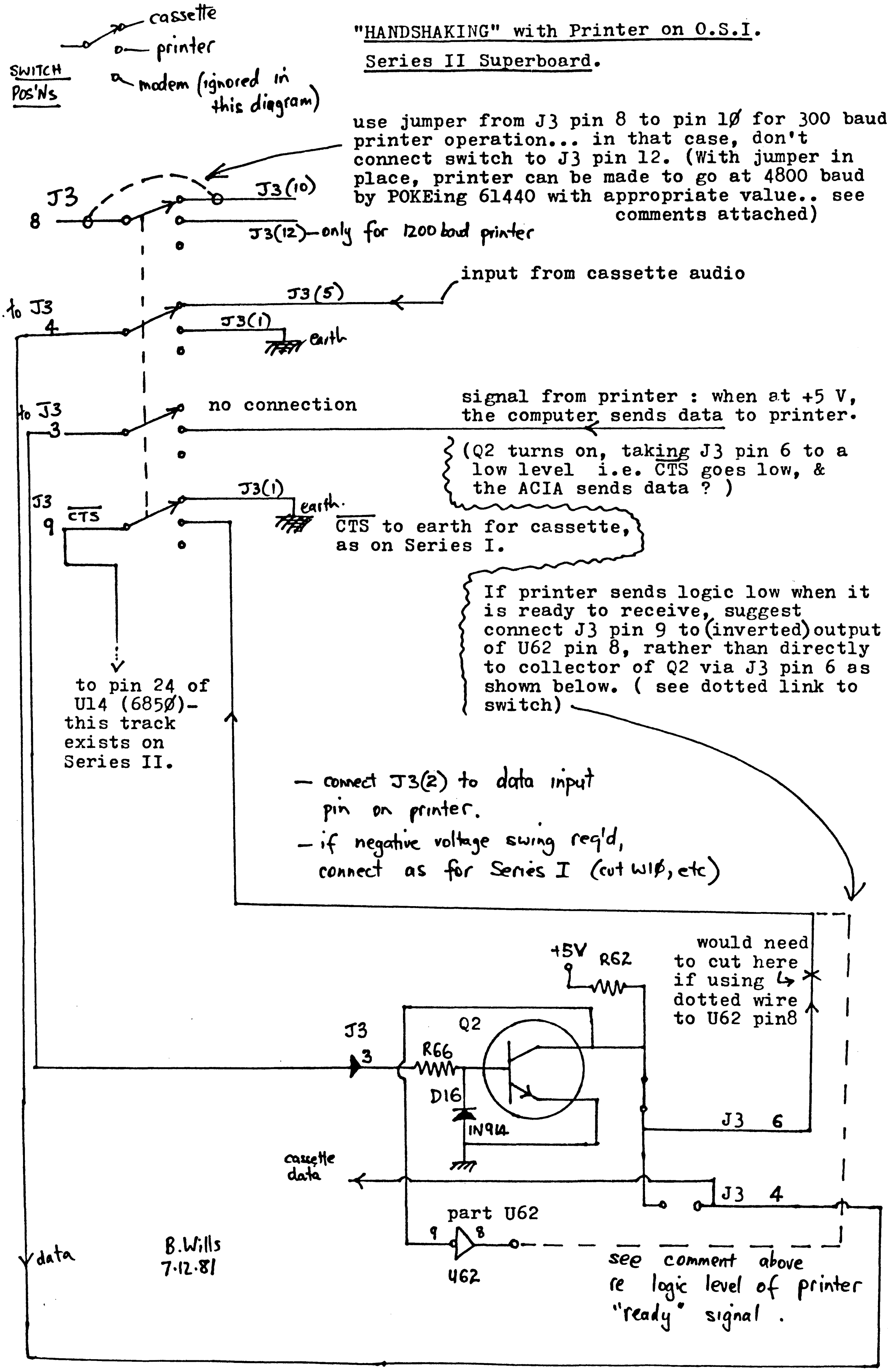
A couple of new games disks have come in, one is 3D Checkers and a game called Protector. Fantastic graphics on Protector, you just need a back-up computer to work the thing out. I will demo them at the next meeting. Well that's it for this month's article, which was written with very short notice as I thought I was going to get a month off.

See you next month,
Gerry

++

"HANDSHAKING" with Printer on O.S.I.

Series II Superboard.



use jumper from J3 pin 8 to pin 1Ø for 300 baud printer operation... in that case, don't connect switch to J3 pin 12. (With jumper in place, printer can be made to go at 4800 baud by POKEing 61440 with appropriate value.. see comments attached)

input from cassette audio

signal from printer : when at +5 V, the computer sends data to printer.

(Q2 turns on, taking J3 pin 6 to a low level i.e. CTS goes low, & the ACIA sends data ?)

CTS to earth for cassette, as on Series I.

If printer sends logic low when it is ready to receive, suggest connect J3 pin 9 to (inverted) output of U62 pin 8, rather than directly to collector of Q2 via J3 pin 6 as shown below. (see dotted link to switch)

to pin 24 of U14 (685Ø)- this track exists on Series II.

- connect J3(2) to data input pin on printer.
- if negative voltage swing req'd, connect as for Series I (cut W1Ø, etc)

would need to cut here if using ↳ dotted wire to U62 pin8

see comment above re logic level of printer "ready" signal.

B. Wills
7·12·81

NOTES ON SERIES II SUPERBOARD RS-232 (also applies to SERIES I)

The notes which came with the S/B show a BAUD RATE switch, which gives 300 baud for cassette and 1200 baud for printer. As I had a lot of problems getting printer going initially, I left J3 pin 8 connected to J3 pin 10. This was how J3 was jumpered when I purchased the S/B in Brisbane. The idea was that 300 baud printer operation might be easier to get going, and not require "handshaking", which I found difficult to follow initially because lack of explanation with the notes supplied.

There is an article in Micro, No. 31, December 1980, p11 called "A C1P User's Notebook", which gives some good info on how the 6850 ACIA works and how the command register may be programmed.

SUMMARY of SOME COMMANDS

For usual format of 8 bit word, no parity and 2 stop bits, and with baud rate set (on switch in diagram or via jumper) at 300;

```
     POKE 61440,3:POKE 61440,17       for normal state.
```

To give 4800 baud without any hardware change (ie. set up as for 300)

```
     POKE 61440,3:POKE 61440,16
```

For format of 7 bit word, odd parity and 1 stop bit, and wiring set as above for 300 baud;

```
     POKE 61440,3:POKE 61440,13       gives 300 baud
     POKE 61440,3:POKE 61440,12       gives 4800 baud
```

The Texas 810 printer works OK at 4800 baud with this last command. It will hold up the computer using the handshaking shown on the previous page when it's buffer is full. This printer required only +5 V for operation, no negative swing.

Bernie Wills

##

This program will convert Machine Code to DATA statements and SAVE them on tape.

```
41000 FORX=1TO32:PRINT:NEXT:PRINT"Machine Code to Basic Data"
41010 PRINT:PRINT"Get blank tape ready"
41020 PRINT:INPUT"Starting address in Hex";A$
41030 GOSUB41200:B$=CHR$(46)+A$+CHR$(47)
41040 S=F:F=0:INPUT"End address in Hex";A$:GOSUB41200
41050 IFF=<STHEN41020
41060 INPUT"Enter the 1st Basic line No.";LN
41070 INPUT"Title of the program";A$
41080 PRINT"Start recorder now":SAVE:FORA=1TO9999:NEXT
41090 PRINTLN;"FORX=1TO32:PRINT:NEXT";CHR$(34);A$
41100 PRINTLN+5;"REM FROM";S;" TO ";F:D=S
41110 LN=LN+10:PRINTLN;"DATA";
41120 PRINTSTR$(PEEK(D));:D=D+1
41130 IFD> FTHENPRINT:GOTO41160
41140 PRINT",";:IFPOS(X) <63THEN41120
41150 PRINTSTR$(PEEK(D):D=D+1:GOTO41120
41160 PRINTLN+10;"FORC=0TO";F-S;":READI:POKE";S;"+C,I:NEXT:END
41170 FORX=1TO9999:NEXT:POKE517,0
41180 PRINT"Recording complete":END
41200 FORX=1TO4:E=ASC(MID$(A$,X,1))
41210 E=E-48:IFE> 16THENE=E-7
41220 F=F*16+E:NEXTX:RETURN:END
```

John Whitehead

MACHINE CODE PROGRAMMING

In response to numerous requests, we are going to try and explain the mysteries of machine code programming. Owing to the different levels of machine code knowledge among members, we will have to start with the basics and ask you members who are more advanced in M/code to be patient until we catch up with you, or better still, any articles or hints you can send us will be much appreciated.

These articles will be written for a C1 with at least 4K of user RAM, (any changes necessary to run the programs on a C2 will be noted at the end of each article). The Assembler/Editor and Extended Monitor are not necessary but will be useful if you know how to use them. As well as your computer, you will need lots of paper (programming sheets for preference), a 6502 mnemonics chart (there is one on page 13 of KAOS volume 1 No.12). a flow chart template (handy but not essential) and any books on 6502 M/code programming.

What we hope to do each month, is introduce and explain the various concepts of M/code programming. The first thing we will do is explain some of the terms we will be using, first the strange word 'mnemonic', this is a shorthand form of an instruction, eg. LDA = Load the Accumulator with memory. Note that your computer does not recognise a mnemonic, you use it to write your program and then either use an Assembler or hand convert it to an OP-CODE.

An Op Code is the Hex form of a mnemonic and is an instruction to the computer to perform an operation, it can have several values depending on the ADDRESS MODE, for instance LDA has 8 Op Codes.

An Addressing Mode is a method by which the computer can access or refer to data. The 6502 uses 13 different addressing modes and we will explain these as they are used in our demo programs.

This month we introduce 2 addressing modes and 4 op codes, the first one is the Absolute Mode, this is a 3 byte instruction and the first byte (the Opcode) tells the computer to read the next 2 bytes as (in this case) it's next operating address. The address is in LO byte - HI byte order.

In the Immediate mode the value of the byte following the Op Code is used 'immediately', as in the CMP instruction where it is compared to the accumulator.

This months program uses:

20 JSR Jump to subroutine and return to the next program line.

4C JMP Jump to the address following the Opcode.

C9 CMP Compare Immediate, subtracts the value following the Opcode from the value in the Accumulator and sets flags according to the result in this case the Z flag will be set to 1 if result is zero and 0 if the result is non-zero.
The N flag will be set to 1 if the result is minus
The C flag will be set to 1 if the value in memory is less than or equal to the Accumulator.

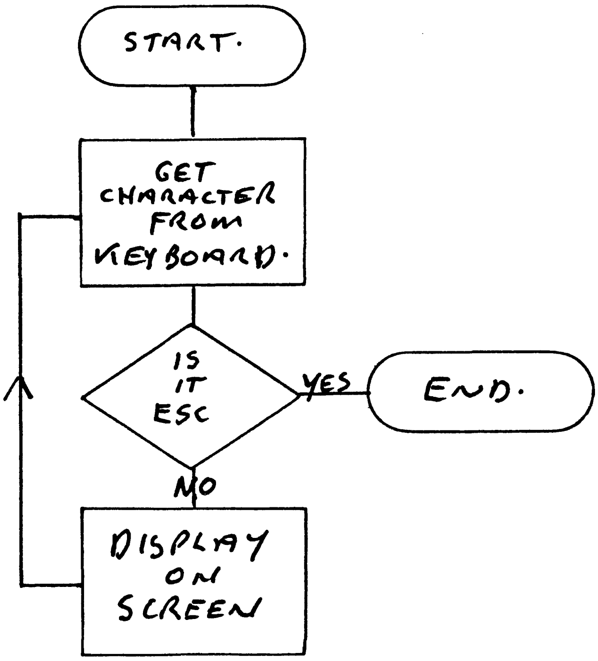
F0 BEQ Branch if equal (to zero). Check Z flag, if flag is 0 drops through to next line and continues program. If flag is 1 then branch to address computed from next data byte.

The secret of writing M/code is to plan (flow chart) your program and use as many existing subroutines as possible, the Monitor and Basic ROM's contain many subroutines that may be used in your programs. eg.

$FEED - Get a character from keyboard
$BF2D - Print a character to the screen
$FE00 - Return to monitor

OSI have very conveniently left a block of unused memory from $0222 to $02FF which is very handy for small programs, especially those that will be accessed by USR(X), as they will not interfer with the Basic program and the area is not cleared by a cold start.

This months program: Object - to display keys pressed and return to monitor when ESC is pressed. The first step is to draw a Flow Chart.



When this is correct start filling in your programming sheet.

Address	Op Code	Label	Mnemonic	Comment
0222		LOOP	JSR $FEED	
			CMP #$1B	
			BEQ END	
			JSR $BF2D	
			JMP LOOP	
		END	JMP MONITOR	

Now put in the opcodes (using the mnemonics list) and the absolute addresses, there is also one forward branch in this program, we will explain how to calculate relative branches later, but for the moment enter it 06

```
0222     20 ED FE   LOOP       JSR $FEED
0225     C9 1B                 CMP #$1B
0227     F0 06                 BEQ END
0229     20 2D BF              JSR $BF2D
022C     4C 22 02              JMP LOOP
022F     4C 00 FE   END        JMP MONITOR
```

Now for the most important part **** COMMENTS ****, these serve the same function as REM's in a BASIC program, they explain what each section of the program is doing, and if you don't put them in, next time you look at the program you won't have a clue what's going on.

```
0222     20 ED FE   LOOP       JSR $FEED     Get character from keyboard
0225     C9 1B                 CMP #$1B      Is it ESC?
0227     F0 06                 BEQ END       No, then END
0229     20 2D BF              JSR $BF2D     Display on screen
022C     4C 22 02              JMP LOOP      Get another character
022F     4C 00 FE   END        JMP MONITOR   Return to Monitor
```

Now you have a program written, the next thing is to enter it into your computer. Switch on and break, then press M, this brings up the address 0000 on the screen. Enter the starting address of your program, 0222 and then press /, the Monitor is now in DATA mode and ready to receive your program. Enter the data with a CR (Return) between each byte, when the data is entered, type .0222/ and use CR to step through the program to make sure it's OK. If there are no mistakes type .0222G and any key you hit after that will be echoed on the screen until you hit ESC which will end the program and jump back to the monitor with 0000 displayed on the screen. To restart your program type 0222G

It is possible to change this program to end on other keys by changing the byte after CMP (at 0225). Find the HEX value of any key (eg H = 48) and enter it at 0228, now run the program and it will end if H is pressed.

As we mentioned earlier, the CMP instruction sets flags according to the result of subtracting memory from accumulator, we can now use these flags to check other Branch instructions, leave 48(H) at 0226 and change the byte at 0227 to the following:

D0	BNE	Branch	if	not equal (to 0)	checks	Z	flag
10	BPL	"	"	plus	"	N	"
30	BMI	"	"	minus	"	N	"
90	BCC	"	"	carry clear	"	C	"
B0	BCS	"	"	carry set	"	C	"

By using more CMP instructions and combinations of the branches it is possible to set up multiple choice situations eg. If the Hex value of the key is pressed is >$41 and <$60 then ....

Next month we will describe the Accumulator and the Op Codes which effect it.

A FAST LOAD/SAVE PROGRAM

The following program started out to be a fast Load/Save for M/Code programs. It will still do this but it will also save Basic programs at a faster than normal rate, it is faster because instead of spelling out the Basic commands it saves the Tokens. This means that the speed increase depends on the amount of text in the program, the more text the less speed. To save a Basic program, BREAK, then type M 0222G. To load a tape, COLD START BREAK M L. When the program is loaded do a WARM START. As the program writes itself on the start of each tape and must be loaded before the program, it is not practical to use it for short programs.

To save M/Code programs put the start address in $79 and $7A and the end address in $7F and $80, both in Lo-Hi format. Then 0222G will start the program and it will write itself to tape then save the main program.

To load the program start the tape and press M L , the first section will load in monitor format, then the program will load, with the byte being loaded flashing on the screen. There is no check on the load but this will not be a problem if you usually don't have trouble loading tapes. This program is three times faster than the Ext. Monitor Load/Save.

The program was written to assemble at $2222 and be relocated to $0222 with the Ext. Monitor, but by changing the bytes at $222A and $2251 to $22 it will run at $2222. If you have a C4 the WRITE address is $BF15 and the READ address is $FBC9.

```
  10 D250=           SCREEN = $D250
  20 FCB1=           WRITE  = $FCB1
  30 FFBF=           READ   = $FFBF
  40 0079=           COUNT  = $79
  50 007F=           END    = $7F
  60 2222            *      = $2222
  70                        ;
  80 2222 A000      PROGRM LDY #0
  90 2224 A92E             LDA #'.        Put monitor in addr. mode
 100 2226 20B1FC           JSR WRITE
 110 2229 A902             LDA #$02       Load start addr of this prog
 120 222B 20DB22           JSR PRT2HX      convert to ASCII and write
 130 222E A922             LDA #$22        to cass.
 140 2230 20DB22           JSR PRT2HX
 150 2233 A92F             LDA #'/        Monitor to data mode
 160 2235 20B1FC           JSR WRITE
 170 2238 B92222    LOOPSB LDA PROGRM,Y   Put READ/WRITE program
 180 223B 8D50D2           STA SCREEN      on cassette
 190 223E 20DB22           JSR PRT2HX
 200 2241 A90D             LDA #$0D       C/R between each byte
 210 2243 20B1FC           JSR WRITE
 220 2246 C8               INY
 230 2247 C0CD             CPY #$CD       CD = length of this program
 240 2249 D0ED             BNE LOOPSB     Loop till done
 250 224B A92E             LDA #'.        Monitor to address mode
 260 224D 20B1FC           JSR WRITE
 270 2250 A902             LDA #$02       Puts 'G' address of READ
 280 2252 20DB22           JSR PRT2HX      program on cass
 290 2255 A996             LDA #LEAD1
 300 2257 20DB22           JSR PRT2HX
 310 225A A947             LDA #'G        Starts READ when re-
 320 225C 20B1FC           JSR WRITE       loading tape
 330                       ;
 340 225F A940      SAVE   LDA #$40       Start of write program
 350 2261 20B1FC           JSR WRITE      Writes 3 @'s as leader
 360 2264 20B1FC           JSR WRITE
 370 2267 20B1FC           JSR WRITE
 380 226A A000             LDY #$00
 390 226C B97900    ADRESS LDA COUNT,Y      Saves addr's required by
 400 226F 20B1FC           JSR WRITE        BASIC program
 410 2272 C8               INY
 420 2273 C008             CPY #$08
 430 2275 D0F5             BNE ADRESS
 440 2277 A000             LDY #$00
 450 2279 B179      LOOP   LDA (COUNT),Y Saves main program, start-
 460 227B 8D50D2           STA SCREEN      ing from addr in $7A-$79
 470 227E 20B1FC           JSR WRITE
 480 2281 E679             INC COUNT
 490 2283 D002             BNE OVRADD
 500 2285 E67A             INC COUNT+1
 510 2287 A579      OVRADD LDA COUNT
 520 2289 C57F             CMP END
 530 228B 90EC             BCC LOOP       If less than, continue
 540 228D A57A             LDA COUNT+1
 550 228F C580             CMP END+1
 560 2291 90E6             BCC LOOP       If less than, continue
 570 2293 4CD022           JMP FINISH
 580                       ;
 590 2296 A000      LEAD1  LDY #$00        Start of read program
 600 2298 20BFFF    LEAD2  JSR READ        Look for 3 @'s
 610 229B C940             CMP #$40
 620 229D D0F7             BNE LEAD1
 630 229F C8               INY
 640 22A0 C003             CPY #$03
 650 22A2 D0F4             BNE LEAD2
 660 22A4 A000             LDY #$00
 670 22A6 20BFFF    RESTOR JSR READ       Load BASIC addresses
 680 22A9 997900           STA COUNT,Y       and store at $79-$80
 690 22AC 8D50D2           STA SCREEN
 700 22AF C8               INY
 710 22B0 C008             CPY #$08
 720 22B2 D0F2             BNE RESTOR
 730 22B4 A000             LDY #$00
 740 22B6 20BFFF    GETBYT JSR READ       Get Byte from ACIA
 750 22B9 9179             STA (COUNT),Y Load main program, start-
 760 22BB 8D50D2           STA SCREEN      ing at addr in $7A-$79
 770 22BE E679             INC COUNT
 780 22C0 D002             BNE OVERAD
 790 22C2 E67A             INC COUNT+1
 800 22C4 A579      OVERAD LDA COUNT
 810 22C6 C57F             CMP END
 820 22C8 90EC             BCC GETBYT     If less than, continue
 830 22CA A57A             LDA COUNT+1
 840 22CC C580             CMP END+1
 850 22CE 90E6             BCC GETBYT     If less than, continue
 860 22D0 A903      FINISH LDA #$03       Restore BASIC start address
 870 22D2 857A             STA COUNT+1
 880 22D4 A901             LDA #$01
 890 22D6 8579             STA COUNT
 900 22D8 4C0000           JMP $0000
 910 22DB 48        PRT2HX PHA            HEX to ASCII routine
 920 22DC 4A               LSR A
 930 22DD 4A               LSR A
 940 22DE 4A               LSR A
 950 22DF 4A               LSR A
 960 22E0 20E422           JSR PRTHEX
 970 22E3 68               PLA
 980 22E4 290F      PRTHEX AND #$0F
 990 22E6 C90A             CMP #$0A
1000 22E8 F8               SED
1010 22E9 6930             ADC #$30
1020 22EB D8               CLD
1030 22EC 4CB1FC           JMP WRITE
```

TYPING TUTOR PROGRAM

```
  5 REM  *Typing Tutor* I031181E  Alter D and LL to suit your screen
 10 FORX=1TO30:PRINT:NEXT
 15 REM   Set up Display
 20 D=53762:LL=32:C=20:D1=D-LL-1:D2=D-3*LL+11:POKE11,0:POKE12,253
 30 FORX=1TOCSTEP2:POKED,221:poked+X,148:POKED+X+1,217:NEXT
 40 POKED+X-1,222:D=D+LL:GOSUB100
 50 POKED+X,222:D=D+LL+1:GOSUB100:IFY=11THENC=18:POKED+X-1,223
 60 GOTO50
100 POKED,149:FORX=1TOCSTEP2:READY:POKED+X,Y+48:POKED+X+1,149:NEXT
110 IFC=18THENGOTO140
120 D=D+LL:POKED,220:FORX=1TOCSTEP2:POKED+X,217:POKED+X+1,215:NEXT
130 RETURN
D=D+LL:POKED,220:FORX=1TOCSTEP2:POKED+X,148:POKED+X+1,215
140 D=D+LL:POKED,220:FORX=1TOCSTEP2:POKED+X,148:POKED+X+1,215
150 NEXT:POKED+X-1,223:POKED+X-2*LL+1,32:FORX=1TO999:NEXT
155 REM  Select key to be blanked
160 C=INT(RND(1)*39+1):RESTORE:FORX=1TOC:READZ:NEXT:Z=Z+48
170 D=D1:FORX=1TO4:D=D+LL*2+1:FORW=1TO20:K=D+W
180 IFPEEK(K)=ZTHEN200
190 NEXTW,X
195 REM  Blank key and check keyboard
200 POKED2,Z:POKEK,32:X=USR(X)
210 IFPEEK(531)=ZTHENPOKEK,187:GOSUB230:POKEK,Z:GOTO160
220 GOSUB230:POKEK,Z:GOSUB230:GOTO200
230 FORX=0TO100:NEXT:RETURN
240 DATA 1,2,3,4,5,6,7,8,9,0,33,39,21,34,36,41,37,25,31,32,17
250 DATA 35,20,22,23,24,26,27,28,11,42,40,19,38,18,30,29,-4,-2
```

+++

DOS line input routine, inputs everything! except C/R.

```
 10 DISK!"GO 0558"
 20 L1$="":FORI=12180 TO 12251
 30 IF PEEK(I)=0 THEN 100
 40 LI$=LI$+CHR$(PEEK(I))
 50 NEXT
100 REM
```

George Nikolaidis

****

LOADING WITHOUT SCREEN LISTING

If listing on the screen when loading a program gives clues away on how to win a game, use the program below to stop screen listing.
To save the main program enter SAVE CR RUN 30100 CR

```
30080 REM   Loading without screen listing
30090 REM   Save (CR) run 30100
30100 PRINT":THIS PROGRAM DOES NOT LIST ON THE SCREEN
30110 PRINT":WAIT FOR THE OK ?SN ERROR THEN
30120 PRINT":PRESS SPACE BAR AND RUN
30130 PRINTCHR$(15):LIST-30000
```

Note the colon in the PRINT statements to stop false SN errors.

John Whitehead